Dowd 3-3

#11

1 of 3

# IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Dr. Patrick W. Dowd et al. Attorney Docket No.: Dowd 3-3
Serial No. 09/287,654 Art Unit: 2131
Filed: April 7, 1999 Examiner: Mr. Christopher A. Revak
For: FIREWALL FOR PROCESSING A CONNECTIONLESS NETWORK PACKET

## APPELLANT'S BRIEF UNDER 37 CFR 1.192

Mail Stop Appeal Brief--Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This brief is in furtherance of the Notice of Appeal filed in the above-identified case on June 30, 2003.

The fees required under § 1.17(f) and any required petition for extension of time for filing this brief and fees therefore are dealt with in the accompanying FEE TRANSMITTAL.

This brief is transmitted in triplicate per 37 CFR 1.192(a).

This brief contains these items under the following headings and in the order set forth below (37 CFR 1.192(C)):

I. REAL PARTY IN INTEREST

II. RELATED APPEALS AND INTERFERENCES

III. STATUS OF CLAIMS

IV. STATUS OF AMENDMENTS

V. SUMMARY OF INVENTION

VI. ISSUES

VII. GROUPINGS OF CLAIMS

VIII. ARGUMENTS

A. REJECTIONS UNDER 35 U.S.C. § 103(a)

IX. APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

The final page of this brief bears the attorney's signature.

## I. REAL PARTY IN INTEREST

The real party in interest in the above-identified appeal is the U.S. Government as represented by the Director, National Security Agency.

## II. RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences related to the above-identified appeal.

## III. STATUS OF CLAIMS

There are a total of 27 claims in the above-identified application. Claim 27 has been canceled. No claims have been withdrawn from consideration but not canceled. Claims 1-26 are pending. Claims 1, 4-8, 14, and 17-21 have been finally rejected. Claims 2, 3, 9-13, 15, 16 and 22-26 were finally objected. Claims 1-26 are appealed.

## IV. STATUS OF AMENDMENTS

There has been no amendment to the above-identified application subsequent to the final rejection of this application.

## V. SUMMARY OF INVENTION

The method of the present invention, as illustrated in Figure 3 and described in the Summary on pages 9-12, is to allow a connectionless network packet access to an information processing network, where the connectionless network packet is compared *only once*, if at all, to a database containing rules for allowing access. A connectionless network packet that is associated with a particular connectionless network flow that is not already pre-approved or pre-disapproved is compared *only once* to the rules for acceptance or rejection. Prior art method compare packets to a database every time. Such prior art methods are inefficient and subject to denial-of-access attackers, where an attacker send sends so many bogus packets that occupy processing time that legitimate packets receive no processing time and, therefore, the system is unavailable for legitimate processing. The present invention is more efficient and is less susceptible to a denial-of-access attack.

The present invention makes an access control determination on the first occurrence of a new connectionless network flow. The connectionless network packet of the new flow initiates the evaluation process. The result of this evaluation is then applied to subsequently received connectionless network packets that are associated with the flow of the evaluated connectionless network packet without having to initiate the evaluation process ever again. Prior art methods initiate an evaluation process for each packet received. Since much computation time is taken up with an exhaustive comparison of a new connectionless network packet against the rules for acceptance or rejection, only having to do this comparison the first time a connectionless network packet having a given set of connectionless network header information is encountered results in a maximally efficient firewall.

The first step of the method is initializing a database, an approved list, and a disapproved list.

The second step is receiving a connectionless network packet.

The third step is computing a flow tag based on the connectionless network packet.

The fourth step is discarding the connectionless network packet and returning to the second step if the flow tag is on the disapproved list.

The fifth step is allowing the connectionless network packet access to the information processing network and returning to the second step if the flow tag is on the approved list.

The sixth step is comparing the flow tag to the database if the flow tag is not on the approved list or the disapproved list.

The seventh step is discarding the connectionless network packet, adding the flow tag to the disapproved list, and returning to the second step if the database rejects the flow tag.

The eighth, and last, step is allowing the connectionless network packet access to the information processing network, adding the flow tag to the approved list, and returning to the second step if the database accepts the flow tag.

The firewall of the present invention minimizes the amount of processing time spent on a subsequently received connectionless network packet if its flow tag was previously approved or rejected as evidenced by the flow tag being on the approved list or the disapproved list. Therefore, additional processing time need not be wasted comparing the flow tag to the database rules ever again. No other firewall is known that performs this "one-touch" approach on a connectionless network packet. The result is fast approvals for approved connectionless network packets and fast disapprovals for disapproved connectionless network packets. Comparison time is only spent on those connectionless network packets having a flow tag that was never encountered before (i.e., the first connectionless network packet of a new flow). Minimizing the time to approve or disapprove a connectionless network packet increases the performance of a firewall and decreases the cost to manufacture such a firewall.

## VI. ISSUES

1. Whether claims 1, 4-8, 14, and 17-21 are obvious in light of an article by Decasper et al (Decasper), entitled "Crossbow—A Toolkit for Integrated Services over Cell Switched IPv6," in view of U.S. Pat. No. 5,826,014 (Coley et al.).

## VII. GROUPINGS OF CLAIMS

Claims 1-26 each stand on their own merit.

## VIII. ARGUMENTS

### A. REJECTIONS UNDER 35 U.S.C. § 103

Examiner rejected Applicants' argument that Examiner is incorrect when he says that a filter disclosed by Decasper is equivalent to Applicants' rules in a database. Applicants respectfully point out that U.S. Pat. No. 5,835,726, which Applicants provided Examiner in their Information Disclosure Statement and included by reference into their patent application, states in the Abstract on the first page that "[a} user generates a rule base which is then converted into a set of filter language instructions"; "[e]ach rule in the rule base includes..."; and "packets are filtered as they flow into and out of the network in accordance with the rules comprising the rule base." Applicants offer that the USPTO has already taken official notice that rules and filters are distinct entities and that just because a filter may be used to implement a rule does not mean that a filter is in fact a rule base. Therefore, Examiner's rejection based on a filter being a rule base must be overturned.

Examiner agreed with Applicants' rebuttal that Decasper does not mention a match for the purpose of determining whether or not to grant access to a packet but then restated his grounds for rejection from the first office action, including the claim that Decasper does include such a match.

Examiner reiterated that he relies on Coley et al. for the use of a disapproval list. Applicants believe that they responded to this point in their first reply, but the Examiner did not respond to their arguments. Instead, Examiner restated his rejections from the first office action. Applicants believe that their response to the first office action is still valid and repeats it below.

Examiner rejected claims 1, 4-8, 14, and 17-21 under 35 U.S.C. §103(a) as being unpatentable over Decasper in view of Coley et al.

As per claim 1 and 14, Examiner said that the Association Identification Unit (AIU) of Decasper stores information pertaining to a flow and filter information or rules. This is not so. Packets received by Decasper are stored in a stack (page 5, line 2). The AIU stores filters not rules (page 5, line 1). When a packet is received by Decasper it is stored in a stack and presented to the filters (page 4, line 13) of the AIU so that the packet may be sent to an appropriate Toolkit Module for further processing (page 4, line 13 and page 5, line 1). Examiner equates the filters to rules. However, the filters do not contain rules but the same fields as the flow with the information in some fields replaced with wildcards (page 4, lines 15-16). Decasper does not include rules for acceptance or rejection as do Applicants. Decasper accepts every packet and makes no judgment concerning acceptability as do Applicants. Decasper merely concerns itself with directing flows to appropriate modules for further processing. Decasper does not determine whether or not a packets should be allowed or denied access as do Applicants. Applicants disclose a method of making access and denial decisions in an efficient manner (page 11, lines 14 to page 12, line 2).

The AIU also associates the packets in a flow with one another by tagging subsequent packets that pertain to a flow with an identifier (page 5, lines 2-3). The first packet in a flow will be unknown. Decasper handles first packets in a flow by automatically accepting it and creating a new flow enter or identifier and mapping the flow to an appropriate packet filter (page 4, lines 4-6) and does not compute anything as Examiner said. The flow identifier is only used to associate packets in a flow so that they get processed by the same filter, not to allow or deny access to further processing as do Applicants (page 20, claim 1), and not as an approval list as Examiner stated. Examiner said that Decasper performs a match and the packet is allowed to pass. Decasper does not mention a match for the purpose of determining whether or not to grant access to a packet.

Examiner said that it is inherent that Decasper initializes the AIU or database. The AIU is not a database but a back of filters with the ability to place an identifier on subsequent packets in a flow.

It is the burden of the U.S. Patent & Trademark Office to establish a prima facie case of obviousness when rejecting claims under 35 U.S.C. §103. In re Reuter, 210 USPQ 249 (CCPA 1981). Decasper is inoperable as a method of making access or denial decisions as do Applicants and, therefore, Decasper is defective for the purpose relied upon by Examiner. In re Hoehsema, 158 USPQ 596 (CCPA 1968). The mere absence from Decasper of an explicit requirement to make access and denial determinations cannot reasonably be construed as an affirmative statement that [the requirement is in the reference]. In re Evanega, 4 USPQ2d 1249. There is no suggestion in Decasper to deny access to a packet. If the prior art does not provide the impetus to do what the inventor had done then Examiner's case of prima facie obviousness fails. In re Herschler, 200 USPQ 711. Examiner said that it would have been obvious to modify Decasper to arrive at Applicants method. However, Decasper does not suggest such modifications. A prior art reference cannot provide the motivation for making a modification if the suggestion for the proposed modification does not come from the prior art reference. In re Gordon, 221 USPQ 1127

(Fed. Cir. 1984). Decasper includes no logical reasoning or justification for modifying their method. There must be some logical reason apparent from the evidence of record that would justify a...modification of references. In re Regel, 188 USPQ 132. Examiner reduces Applicants' method to the idea of allowing or denying access to a packet. The invention cannot be tested on the basis of whether the "idea" is patentable. Under the patent statute, Title 35 U.S.C., "ideas" are not patentable; claimed structures and methods are. Reducing a claimed invention to an "idea," and then determining patentability of that "idea" is error. Analysis properly begins with the claims, for they measure and define the invention. The "difference" may have seemed slight (as has often been the case with some of history's great inventions, e.g., the telephone), but it may also have been the key to success and advancement in the art resulting from the invention. Further, it is irrelevant in determining obviousness that all or all other aspects of the claim may have been well known in the art. Jones v. Hardy, 220 USPQ 1021 (Fed. Cir. 1984). At best, in view of these disclosures, one skilled in the art might find it obvious to try various...[modifications]... However, this is not the standard of 35 U.S.C. section 103. In re Goodwin, 198 USPQ 1. Proper claim construction demands interpretation of the entire claim in context, not a single element in isolation. Hockerson-Halberstadt, Inc. v. Converse, Inc., Fed. Cir. No. 98-1501, 7/20/99. Differences between an invention and the prior art cited against it cannot be ignored...Under section 103, [Examiner] cannot dissect a claim, excise [a claim limitation] from it, and declare the remaining portion of the mutilated claim to be unpatentable. The claims must be read as a whole. If [Examiner] meant to disregard that basic principle of claim interpretation, [the court] must reverse as a matter of law. In re Gulack, 217 USPQ 401 (Fed. Cir. 1983). One important indicium of non-obviousness is "teaching away" from the claimed invention by the prior art...at (and/or after) the time the invention was made. U.S. v. Adams, 148

USPQ 479 (1966). Decasper discloses a method of steering packets and not a method of determining acceptability of a packet as do Applicants and, therefore, teaches away from Applicants' method.

Examiner admits that Decasper does not disclose a disapproved list. However, Examiner said that Coley et al. discloses a disapprove list. Coley requires a series of tests for determining accessibility (Fig. 4A and Fig. 4B). Having to execute a series of tests for each packet is very inefficient. Applicants' method is an improvement over such an inefficient system. Instead of executing a series of tests on every packet presented to it, Applicants initialize a database with approved and disapproved flow tags that are used to determine accessibility (page 20, claim 1), not identification of one packet with another of the same flow. Applicants also compute a flow tag, not a flow identifier, for any flow that is not pre-approved or pre-disapproved. So, Applicants' method only does a computation when it is presented with a new flow and not for every packet presented to it (page 11, line 14 to page 12, line 2). Therefore, Applicants' method is more efficient than that of Coley et al.

Examiner said that it would have been obvious to add a means for discarding a packet. This is not what Applicants have done. Applicants have invented an efficient method of allowing or denying access to a computer that does not require each and every packet presented thereto to be tested. This improvement is not found in Coley et al., Decasper or the combination thereof.

It is the burden of the U.S. Patent & Trademark Office to establish a prima facie case of obviousness when rejecting claims under 35 U.S.C. §103. In re Reuter, 210 USPQ 249 (CCPA 1981). Coley et al. is inoperable as a method of allowing or denying access where tests are made only for never-before-seen packets as Applicants' method and, therefore, Coley et al. is defective for the purpose relied upon by Examiner. In re Hoehsema, 158 USPQ 596 (CCPA 1968). Neither Decasper nor Coley et al. suggests the combination or modification suggested by Examiner.

The method Examiner tries to create by combining Decasper and Coley et al. only comes from Applicants' specification. Examiner may not combine references if the suggestion to combine the references comes from Applicant's own specification. In re Jansson, 203 USPQ 976 (CCPA 1979). The combination and modifications suggested by Examiner destroys the intended function of both Decasper and Coley et al. References are not properly combinable or modifiable if their intended function is destroyed. In re Gordon, 221 USPQ 1125(fed. Cir. 1984).

Neither Decasper nor Coley et al. include any reasoning for the combination and modifications suggested by Examiner. There must be some logical reason apparent from the evidence of record that would justify a...modification of references. In re Regel, 188 USPQ 132.

A prior art reference cannot provide the motivation for making a modification if the suggestion for the proposed modification does not come from the prior art reference. In re Gordon, 221 USPQ 1127 (Fed. Cir. 1984). At best, in view of these disclosures, one skilled in the art might find it obvious to try various...[modifications]...However, this is not the standard of 35 U.S.C. section 103. In re Goodwin, 198 USPQ 1. Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching, suggestion, or incentive supporting the combination. ACS Hospital Systems, Inc. v. Montefiore Hospital, 221 USPQ 929 (Fed. Cir. 1984). Neither Decsaper nor Coley et al. includes such teaching, suggestion, or incentive. If the prior art does not provide the impetus to do what the inventor had done then Examiner's case of prima facie obviousness fails. In re Herschler, 200 USPQ 711.

Even if all elements of a claimed invention are disclosed in various prior art references, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill would have been prompted to combine the teachings of the references to arrive at the claimed invention. In re Regel, 188 USPQ 132 (CCPA 1975).

Examiner has not found such a reason in the prior art. Examiner merely said that it would have been obvious to modify the prior art references to obtain Applicants' method. This is not the standard for determining obviousness.

As per claims 4 and 17, Examiner said that Coley et al. discloses the processing of IP packets. Claims 4 and 17 are dependent on claims 1 and 14, respectively, and contain all of the limitations of the claims on which they depend. Therefore, for the reasons outlined above, Applicants submit that Examiner's rejections of claims 1 and 14 are improper and, therefore, Examiner's rejections of claims 4 and 17 are improper as well.

As per claims 5 and 18, Examiner said that Decasper discloses receiving packets associated with an identifier, creating an identifier for each new flow, storing the new identifier, and allowing the packet access to the system. As outlined above, Decasper uses as identifier to associate packets within a flow so that such packets may be presented to the same filter so that all of the packets in a flow are processes by the same Toolkit Module and does not use the identifier to make any determination as to accessibility of the packet. Decasper grant access to all packets. Furthermore, claims 5 and 18 are dependent on claims 1 and 14, respectively, and contain all of the limitations of the claims on which they depend. Therefore, for the reasons outlined above, Applicants submit that Examiner's rejections of claims 1 and 14 are improper and, therefore, Examiner's rejections of claims 5 and 18 are improper as well.

As per claims 6 and 19, Examiner admits that neither Decasper nor Coley et al. disclose recording each allowed access. However, Examiner took official notice that such a concept is obvious and that it would have been obvious to do so. Concepts are not patentable. Applicants have not proposed a concept but a combination of steps, which is patentable subject matter. Claims 6 and 19 are dependent on claims 1 and 14, respectively, and contain all of the limitations of the claims on which they depend. Therefore, for the reasons outlined above, Applicants submit that Examiner's rejections of claims 1 and 14 are improper and, therefore, Examiner's rejections of claims 6 and 19 are improper as well. Furthermore, would have been obvious is not the standard. (cite).

As per claims 7, 8, 20, and 21, Examiner admits that neither Decasper nor Coley et al. disclose alerting a system administrator if the number of discarded packets exceeds a user-definable threshold within a user-definable span of time. However, Examiner took official notice that such a concept is obvious and that it would have been obvious to do so. Concepts are not patentable. Applicants have not proposed a concept but a combination of steps, which is patentable subject matter. Claims 7, 8, 20, and 21 are dependent on claims 1 and 6, and claims 14 and 19, respectively, and contain all of the limitations of the claims on which they depend. Therefore, for the reasons outlined above, Applicants submit that Examiner's rejections of claims 1, 6, 14, and 19 are improper and, therefore, Examiner's rejections of claims 7, 8, 20, and 21 are improper as well.

## IX. APPENDIX

1. A method of accessing an information processing network, comprising the steps of:

a) initializing a database, an approved list, and a disapproved list, where the database contains rules for allowing and denying access to the information processing network, where the approved list includes approvals of connectionless network packets, and where the disapproved list includes disapprovals of connectionless network packets;

b) receiving a connectionless network packet;

c) computing a flow tag based on the connectionless network packet;

d) discarding the connectionless network packet and returning to step (b) if the flow tag is on the disapproved list;

e) allowing the connectionless network packet access to the information processing network and returning to step (b) if the flow tag is on the approved list;

f) comparing the flow tag to the database if the flow tag is not on the approved list and is not on the disapproved list;

g) discarding the connectionless network packet, adding the flow tag to the disapproved list, and returning to step (b) if the database rejects the flow tag; and

h) allowing the connectionless network packet access to the information processing network, adding the flow tag to the approved list, and returning to step (b) if the database accepts the flow tag.

2. The method of claim 1, wherein said step of computing a flow tag is comprised of the steps of:

a) extracting from the connectionless network packet a user-definable number of bits from a connectionless network source address, a connectionless network destination address, a connectionless network protocol, an upper layer protocol header if included in the connectionless network packet, and application layer data;

b) substituting all zeros for the upper layer protocol layer if none is included in the connectionless network packet;

c) setting a user-definable number and location of bits in the result of the last step to zero; and

d) computing a flow tag address.

3. The method of claim 2, where said step of computing a flow tag address is comprised of the steps of:

a) setting a zeroth bit of the flow tag address to $f_0 = s_0 x s_{14} x s_{28} x d_{13} x d_{27} x h_0 x h_{16}$, where x is a bitwise exclusive-or operation, $f_i$ is the ith bit of the flow tag address, where $s_i$ is the ith bit of a connectionless network source address, where $d_i$ is the ith bit of a connectionless network destination address, where $p_i$ is the ith bit of a connectionless network protocol, and where $h_i$ is the ith bit of the upper layer protocol header, and;

b) setting a first bit of the flow tag address to $f_1 = s_1 x s_{15} x s_{29} x d_{12} x d_{26} x h_1 x h_{17}$;

c) setting a second bit of the flow tag address to $f_2 = s_2 x s_{16} x s_{30} x d_{11} x d_{25} x h_2 x h_{18} x p_0$;

d) setting a third bit of the flow tag address to $f_3 = s_3 x s_{17} x s_{31} x d_{10} x d_{24} x h_3 x h_{19} x p_1$;

e) setting a fourth bit of the flow tag address to $f_4 = s_4 x s_{18} x d_9 x d_{23} x h_4 x h_{20} x p_2$;

f) setting a fifth bit of the flow tag address to $f_5 = s_5 x s_{19} x d_8 x d_{22} x h_5 x h_{21} x p_3$;

g) setting a sixth bit of the flow tag address to $f_6 = s_6 x s_{20} x d_7 x d_{21} x h_6 x h_{22} x h_{28} x p_4$;

h) setting a seventh bit of the flow tag address to $f_7 = s_7 x s_{21} x d_6 x d_{20} x h_7 x h_{23} x h_{29} x p_5$;

i) setting a eighth bit of the flow tag address to $f_8 = s_8 x s_{22} x d_5 x d_{19} x h_8 x h_{24} x h_{30} x p_6$;

j) setting a ninth bit of the flow tag address to $f_9 = s_9 x s_{23} x d_4 x d_{18} x h_9 x h_{25} x h_{31} x p_7$;

k) setting a tenth bit of the flow tag address to $f_{10} = s_{10} x s_{24} x d_3 x d_{17} x d_{31} x h_{10} x h_{26}$;

l) setting a eleventh bit of the flow tag address to $f_{11} = s_{11} x s_{25} x d_2 x d_{16} x d_{30} x h_{11} x h_{27}$;

m) setting a twelfth bit of the flow tag address to $f_{12} = s_{12} x s_{26} x d_1 x d_{15} x d_{29} x h_{12} x h_{14}$; and

n) setting a thirteenth bit of the flow tag address to $f_{13} = s_{13} x s_{27} x d_0 x d_{14} x d_{28} x h_{13} x h_{15}$.

4. The method claim 1, wherein the step of discarding the connectionless network packet, adding the flow tag to the disapproved list, and returning to step (b) if the database rejects the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the disapproved list at the flow tag address;

b) determining that the flow tag is on the disapproved list if a match occurred in the last step;

c) discarding the connectionless network packet;

d) adding the flow tag to the disapproved list; and

e) returning to step (b).

5. The method claim 1, wherein the step of allowing the connectionless network packet access to the information processing network, adding the flow tag to the approved list, and returning to step (b) if the database accepts the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the approved list at the flow tag address;

b) determining that the flow tag is on the approved list if a match occurred in the last step;

c) allowing the connectionless network packet access to the information processing network;

d) adding the flow tag to the approved list; and

e) returning to step (b).

6. The method of claim 1, further including the step of recording all allowances of access to the information processing network and recording all discarded connectionless network packets.

7. The method of claim 6, further including the step of alerting a system administrator if the number of discarded connectionless network packets exceed a user-definable threshold.

8. The method of claim 6, further including the step of alerting a system administrator if the number of discarded connectionless network packets exceed a user-definable threshold within a user-definable span of time.

9. The method claim 3, wherein the step of discarding the connectionless network packet, adding the flow tag to the disapproved list, and returning to step (b) if the database rejects the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the disapproved list at the flow tag address;

b) determining that the flow tag is on the disapproved list if a match occurred in the last step;

c) discarding the connectionless network packet;

d) adding the flow tag to the disapproved list; and

e) returning to step (b).

10. The method claim 9, wherein the step of allowing the connectionless network packet access to the information processing network, adding the flow tag to the approved list, and returning to step (b) if the database accepts the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the approved list at the flow tag address;

b) determining that the flow tag is on the approved list if a match occurred in the last step;

c) allowing the connectionless network packet access to the information processing network;

d) adding the flow tag to the approved list; and

e) returning to step (b).

11. The method of claim 10, further including the step of recording all allowances of access to the information processing network and recording all discarded connectionless network packets.

12. The method of claim 11, further including the step of alerting a system administrator if the number of discarded connectionless network packets exceed a user-definable threshold.

13. The method of claim 11, further including the step of alerting a system administrator if the number of discarded connectionless network packets exceed a user-definable threshold within a user-definable span of time.

14. A method of accessing an information processing network, comprising the steps of:

a) initializing a database, an approved list, and a disapproved list, where the database contains rules for allowing and denying access to the information processing network, where the approved list includes approvals of IP packets, and where the disapproved list includes disapprovals of IP packets;

b) receiving an IP packet;

c) computing a flow tag based on the IP packet;

d) discarding the IP packet and returning to step (b) if the flow tag is on the disapproved list;

e) allowing the IP packet access to the information processing network and returning to step (b) if the flow tag is on the approved list;

f) comparing the flow tag to the database if the flow tag is not on the approved list and is not on the disapproved list;

g) discarding the IP packet, adding the flow tag to the disapproved list, and returning to step (b) if the database rejects the flow tag; and

h) allowing the IP packet access to the information processing network, adding the flow tag to the approved list, and returning to step (b) if the database accepts the flow tag.

15. The method of claim 14, wherein said step of computing a flow tag is comprised of the steps of:

a) extracting from the IP packet a user-definable number of bits from a IP source address, a IP destination address, a IP protocol, an upper layer protocol header if included in the IP packet, and data;

b) substituting all zeros for the upper layer protocol layer if none is included in the IP packet;

c) setting a user-definable number and location of bits in the result of the last step to zero; and

d) computing a flow tag address.

16. The method of claim 15, where said step of computing a flow tag address is comprised of the steps of:

a) setting a zeroth bit of the flow tag address to $f_0 = s_0 \text{ x } s_{14} \text{ x } s_{28} \text{ x } d_{13} \text{ x } d_{27} \text{ x } h_0 \text{ x } h_{16}$, where x is a bitwise exclusive-or operation, $f_i$ is the ith bit of the flow tag address, where $s_i$ is the ith bit of a IP source address, where $d_i$ is the ith bit of a IP destination address, where $p_i$ is the ith bit of a IP protocol, and where $h_i$ is the ith bit of the upper layer protocol header, and;

b) setting a first bit of the flow tag address to $f_1 = s_1 \text{ x } s_{15} \text{ x } s_{29} \text{ x } d_{12} \text{ x } d_{26} \text{ x } h_1 \text{ x } h_{17}$;

c) setting a second bit of the flow tag address to $f_2 = s_2 \text{ x } s_{16} \text{ x } s_{30} \text{ x } d_{11} \text{ x } d_{25} \text{ x } h_2 \text{ x } h_{18} \text{ x } p_0$;

d) setting a third bit of the flow tag address to $f_3 = s_3 \text{ x } s_{17} \text{ x } s_{31} \text{ x } d_{10} \text{ x } d_{24} \text{ x } h_3 \text{ x } h_{19} \text{ x } p_1$;

e) setting a fourth bit of the flow tag address to $f_4 = s_4 \text{ x } s_{18} \text{ x } d_9 \text{ x } d_{23} \text{ x } h_4 \text{ x } h_{20} \text{ x } p_2$;

f) setting a fifth bit of the flow tag address to $f_5 = s_5 \text{ x } s_{19} \text{ x } d_8 \text{ x } d_{22} \text{ x } h_5 \text{ x } h_{21} \text{ x } p_3$;

g) setting a sixth bit of the flow tag address to $f_6 = s_6 \text{ x } s_{20} \text{ x } d_7 \text{ x } d_{21} \text{ x } h_6 \text{ x } h_{22} \text{ x } h_{28} \text{ x } p_4$;

h) setting a seventh bit of the flow tag address to $f_7 = s_7 \text{ x } s_{21} \text{ x } d_6 \text{ x } d_{20} \text{ x } h_7 \text{ x } h_{23} \text{ x } h_{29} \text{ x } p_5$;

i) setting a eighth bit of the flow tag address to $f_8 = s_8 \text{ x } s_{22} \text{ x } d_5 \text{ x } d_{19} \text{ x } h_8 \text{ x } h_{24} \text{ x } h_{30} \text{ x } p_6$;

j) setting a ninth bit of the flow tag address to $f_9 = s_9 \text{ x } s_{23} \text{ x } d_4 \text{ x } d_{18} \text{ x } h_9 \text{ x } h_{25} \text{ x } h_{31} \text{ x } p_7$;

k) setting a tenth bit of the flow tag address to $f_{10} = s_{10} \text{ x } s_{24} \text{ x } d_3 \text{ x } d_{17} \text{ x } d_{31} \text{ x } h_{10} \text{ x } h_{26}$;

l) setting a eleventh bit of the flow tag address to $f_{11} = s_{11} \text{ x } s_{25} \text{ x } d_2 \text{ x } d_{16} \text{ x } d_{30} \text{ x } h_{11} \text{ x } h_{27}$;

m) setting a twelfth bit of the flow tag address to $f_{12} = s_{12} \text{ x } s_{26} \text{ x } d_1 \text{ x } d_{15} \text{ x } d_{29} \text{ x } h_{12} \text{ x } h_{14}$; and

n) setting a thirteenth bit of the flow tag address to $f_{13} = s_{13} \text{ x } s_{27} \text{ x } d_0 \text{ x } d_{14} \text{ x } d_{28} \text{ x } h_{13} \text{ x } h_{15}$.

17. The method claim 14, wherein the step of discarding the IP packet, adding the flow tag to the disapproved list, and returning to step (b) if the database rejects the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the disapproved list at the flow tag address;

b) determining that the flow tag is on the disapproved list if a match occurred in the last step;

c) discarding the IP packet;

d) adding the flow tag to the disapproved list; and

e) returning to step (b).

18. The method claim 14, wherein the step of allowing the IP packet access to the information processing network, adding the flow tag to the approved list, and returning to step (b) if the database accepts the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the approved list at the flow tag address;

b) determining that the flow tag is on the approved list if a match occurred in the last step;

c) allowing the IP packet access to the information processing network;

d) adding the flow tag to the approved list; and

e) returning to step (b).

19. The method of claim 14, further including the step of recording all allowances of access to the information processing network and recording all discarded IP packets.

20. The method of claim 19, further including the step of alerting a system administrator if the number of discarded IP packets exceed a user-definable threshold.

21. The method of claim 19, further including the step of alerting a system administrator if the number of discarded IP packets exceed a user-definable threshold within a user-definable span of time.

22. The method claim 16, wherein the step of discarding the IP packet, adding the flow tag to the disapproved list, and returning to step (b) if the database rejects the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the disapproved list at the flow tag address;

b) determining that the flow tag is on the disapproved list if a match occurred in the last step;

c) discarding the IP packet;

d) adding the flow tag to the disapproved list; and

e) returning to step (b).

23. The method claim 22, wherein the step of allowing the IP packet access to the information processing network, adding the flow tag to the approved list, and returning to step (b) if the database accepts the flow tag is comprised of the steps of:

a) comparing the flow tag to any data stored on the approved list at the flow tag address;

b) determining that the flow tag is on the approved list if a match occurred in the last step;

c) allowing the IP packet access to the information processing network;

d) adding the flow tag to the approved list; and

e) returning to step (b).

24. The method of claim 23, further including the step of recording all allowances of access to the information processing network and recording all discarded IP packets.

25. The method of claim 24, further including the step of alerting a system administrator if the number of discarded IP packets exceed a user-definable threshold.

26. The method of claim 24, further including the step of alerting a system administrator if the number of discarded IP packets exceed a user-definable threshold within a user-definable span of time.

Respectfully submitted,

Robert D. Morelli<br>
Reg. No. #37,398

ATTN: AGC(IP&T)
National Security Agency
9800 Savage Road, STE 6542
FT. Meade, MD 20755-6542

(301) 688-0287